A Novel Ensemble Learning Approach to Unsupervised Record Linkage

Anna Jurek, Jun Hong, Yuan Chi, Weiru Liu

PII: S0306-4379(16)30506-3
DOI: 10.1016/j.is.2017.06.006
Reference: IS 1228

To appear in: *Information Systems*

Received date: 21 October 2016
Revised date: 25 June 2017
Accepted date: 27 June 2017

Please cite this article as: Anna Jurek, Jun Hong, Yuan Chi, Weiru Liu, A Novel Ensemble Learning Approach to Unsupervised Record Linkage, *Information Systems* (2017), doi: 10.1016/j.is.2017.06.006

Highlights

- A novel unsupervised approach to record linkage has been proposed
- The approach combines ensemble learning and automatic self learning
- An ensemble of diverse self learning models is generated through application of different string similarity metrics schemes
- Application of ensemble learning alleviates the problem of having to select the most suitable similarity metric scheme and improves the performance of an individual self learning model
- The proposed method obtained comparable results with the supervised methods

A Novel Ensemble Learning Approach to Unsupervised Record Linkage

Anna Jurek[a], Jun Hong[b], Yuan Chi[a], Weiru Liu[c]

[a]*School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Computer Science Building, 18 Malone Road, BT9 5BN Belfast, United Kingdom*
[b]*Department of Computer Science and Creative Technologies, University of the West of England, Coldharbour Lane, BS16 1QY Bristol, United Kingdom*
[c]*Merchant Venturers School of Engineering, University of Bristol,75 Woodland Road, BS8 1UB Bristol, United Kingdom*

Abstract

Record linkage is a process of identifying records that refer to the same real-world entity. Many existing approaches to record linkage apply supervised machine learning techniques to generate a classification model that classifies a pair of records as either match or non-match. The main requirement of such an approach is a labelled training dataset. In many real-world applications no labelled dataset is available hence manual labelling is required to create a sufficiently sized training dataset for a supervised machine learning algorithm. Semi-supervised machine learning techniques, such as self-learning or active learning, which require only a small manually labelled training dataset have been applied to record linkage. These techniques reduce the requirement on the manual labelling of the training dataset. However, they have yet to achieve a level of accuracy similar to that of supervised learning techniques. In this paper we propose a new approach to unsupervised record linkage based on a combination of ensemble learning and enhanced automatic self-learning. In the proposed approach an ensemble of automatic self-learning models is generated with different similarity measure schemes. In order to further improve the automatic self-learning process we incorporate field weighting into the automatic seed selection for each of the self-learning models. We propose an unsupervised diversity measure to ensure that there is high diversity among the selected self-learning models. Finally, we propose to use the contribution ratios of self-learning mod-

els to remove those with poor accuracy from the ensemble. We have evaluated our approach on 4 publicly available datasets which are commonly used in the record linkage community. Our experimental results show that our proposed approach has advantages over the state-of-the-art semi-supervised and unsupervised record linkage techniques. In 3 out of 4 datasets it also achieves comparable results to those of the supervised approaches.

Keywords: Unsupervised record linkage, data matching, classification, ensemble learning

1. Introduction

Record linkage (RL), also referred to as data matching or entity resolution, is a process of finding records that correspond to the same entity from one or more data source [1]. Given two data sources, each pair of records from the data sources can be classified into one of two classes: match and non-match. Table 1 shows a simple example of RL. The table contains records from two bibliographic data sources (DBLP, ACM digital library). The aim is to identify those pairs of records referring to the same publications, which in this case are (ACM1, DB1) and (ACM2, DB2). Any other pairs of records should be identified as non-match. RL has been widely applied in data management, data warehousing, business intelligence, historical data collection and medical research [2]. If records have error-free and unique identifiers, such as social security numbers, RL is a straightforward process that can be easily performed by the standard database join operation. In many cases, however, such a unique identifier does not exist and the linkage process needs to be performed by matching the corresponding fields of two records. A lot of efforts have been made to develop techniques for RL [3]. Unfortunately, in many cases the same data can be represented in different ways in different data sources due to factors such as different conventions (e.g., International Conference on Management of Data versus SIGMOD Conference). Furthermore, data quality is often affected by typographical errors, missing and out of date values. As a consequence, it is

Table 1: An Example of RL.

ID	Authors	Title	Venue
ACM1	A compact B-tree	Peter Bumbulis, Ivan T. Bowman	International Conference on Management of Data
ACM2	A theory of redo recovery	David Lomet, Mark Tuttle	International Conference on Management of Data
DB1	A compact B-tree	Ivan T. Bowman, Peter Bumbulis	SIGMOD Conference
DB2	A theory of redo recovery	Mark R. Tuttle, David B. Lomet	SIGMOD Conference

not always the case that records referring to the same entity have the same values on the corresponding fields. Therefore, sophisticated RL techniques are required in order to identify records that refer to the same real-world entity.

Existing RL methods rely strongly on use of similarity measure, in which selection of such measures is a key issue [3]. There are many of available similarity measures, e.g., Edit Distance, Jaro and Smith-Waterman [3]. Depending on the types of data in RL, similarity measures have different levels of accuracy [4].

Linking two large datasets could be computationally expensive. If we need to link two datasets, A and B, then potentially we should compare each record from A with each record from B. Hence, the total number of comparisons would be $|A|\times|B|$. To reduce the potentially large number of record comparisons different forms of indexing and filtering, collectively referred to as blocking [5], [6], are deployed in RL systems. The idea of blocking is to use a blocking function to divide records into a set of blocks. The candidate pairs of records for linkage are then selected from records in the same block only. The existing approaches to RL can be broadly divided into two categories. The first category of approaches rely on applying generic rules and similarity measures to identify those pairs of records that are similar enough to be matched [7], [8]. The second category

of approaches rely on a training dataset to train a classification model using appropriate statistical and machine learning techniques. In these techniques, each pair of records is represented as a similarity vector with N elements. The similarity vector represents a set of N numeric similarities, each calculated with a similarity measure on the corresponding pair of field values of the two records. The task of RL is then considered as a similarity vector classification problem [9], i.e., whether a similarity vector is classified as match or non-match, for a pair of matching or non-matching records respectively.

The main limitation of the second category of approaches is that they require a labelled training dataset. In many real world situations no labelled dataset is available. In addition, given the sensitivity of the data in RL, very often it is not possible to manually label the records [10]. Therefore, there is a big need for fully unsupervised RL models. To meet this challenge we propose a new approach to RL, which does not require any labelled dataset. The proposed approach combines ensemble learning [11] and automatic self-learning techniques [12]. With self-learning [13] a supervised learning algorithm is trained on a small set of labelled records (seeds) to generate an initial classification model. The initial classifier is then applied to unlabelled records in order to generate more labelled records for the supervised learning algorithm. Only those records that the classifier is most certain about are labelled and added into the training dataset. The updated training dataset is further used to generate another classifier. This process iterates until all unlabelled records have been labelled with the final classifier generated. When there is no labelled record readily available for training the initial classifier, a small set of records needs to be selected and labelled automatically (automatic seed selection). It has been shown that self-learning can outperform other unsupervised algorithms, such as Expectation-Maximization [13]. Self-learning with automatic seed selection has been already investigated in RL and promising results have been achieved [12].

The goal of ensemble learning [11], [14] is to train and combine a number of different classification models to obtain better performance than any of those individual classifiers. A combination of multiple classification models is refereed

to as a classifier ensemble. Classification models included in an ensemble are commonly refereed to as base classifiers (BCs). Ensemble learning has been shown to improve the performance of classification algorithms, such as SVM [15], Decision Tree [16] and Nearest-Neighbor [17]. The key objective in ensemble learning is to train a set of highly diverse base classifiers. Two classification models are considered highly diverse if they make mistakes on (misclassify) different groups of instances. It has been shown that combining classifiers with low diversity does not improve classification accuracy [18]. One approach to training a collection of diverse classification models is to use a different (randomly selected) subset of the training dataset to train each of the classifiers (Bagging) [19]. Classifiers trained on different datasets are expected to make different classification decisions. An alternative is to, instead of randomly sampling from the training dataset, randomly select features, like in the case of Random Forest [20]. With Random Forest, each of the BC is trained with a different subset of features. With another technique, referred to as Boosting [21], the weights of the instances from the training set are dynamically altered based on the accuracy of the classifier. After a BC is built and added to the ensemble, all instances are re-weighted: those that have been correctly or incorrectly classified lose or gain weights respectively. The modified distribution of the training dataset is taken under consideration in the training process of the next BC. Any supervised classification learning algorithm can be applied with the aforementioned techniques to generate the base classifiers. The goal of all the ensemble techniques is to construct a multitude of BCs at training time and output the class that is the mode of their individual predictions at classification time. Very often only a subset of BCs is selected in order to increase the diversity of the ensemble. A common practice is to measure diversity among a group of BCs and select a group of classifiers with the highest diversity to form an ensemble [22]. A number of methods for measuring the diversity among the BCs in an ensemble have been proposed, including pairwise and non-pairwise diversity measures [23]. Pairwise diversity measures are defined for a pair of BCs and the diversity of the ensemble is obtained by averaging all the pairwise

diversity values. Non-pairwise diversity measures are defined on the ensemble as a whole.

Our proposed approach incorporates ensemble learning and self-learning techniques into RL. Self-learning with automatic seed selection addresses the problem of lack of labelled datasets. However, it has yet to achieve a level of accuracy similar to that of supervised learning techniques [12]. Another issue with the unsupervised models for RL is related to the selection of an appropriate similarity measure used for generating similarity vectors. The performance of the RL algorithms relies strongly on the selected similarity measure [4]. It is, however, difficult to select an appropriate similarity measure without labelled datasets [24]. With our proposed approach we generate an ensemble of diverse self-learning models by applying different combinations of similarity measure. By using different combinations of similarity measures we can generate different sets of similarity vectors that can be used to generate different self-learning models. To ensure high diversity among the self-learning models we apply the proposed seed Q-statistic diversity measure. We also propose to use Contribution Ratios of BCs to eliminate those with very poor accuracy from the final ensemble.

In this paper we make the following contributions: (1) We propose a framework for unsupervised RL, which incorporates ensemble learning and self-learning into RL. (2) We improve the existing automatic self-learning technique for RL [12] by incorporating unsupervised field weighting into the process of automatic selection of seeds. (3) We propose an ensemble learning method based on the selection of different similarity measure schemes. This alleviates the problem of having to select the most suitable similarity measure scheme and improves the performance of an individual classifier. (3) We propose a new unsupervised diversity measure, which ensures high diversity among the self-learning models. (4) We propose to use the contribution ratios of BCs to eliminate the weakest BCs from the final ensemble.

2. Related Work

Given the importance and challenges of RL, there has been strong interest in the last decade and significant progress has been made in this field [25][26]. In order to address the problem of lack of labelled data various semi-supervised learning techniques have been proposed for RL over the past few years. In semi-supervised learning only a small set of labelled instances and a large set of unlabelled instances are used in the training process. A popular approach to semi-supervised RL is referred to as active learning (AL) [27]. AL identifies highly informative instances for manual labelling that are later used for training classification models. In [28] the most informative instances are selected for manual labelling based on the classifications by a group of classifiers. The instances that are not assigned to the same class by majority of the classifiers are selected for manual labelling. In [29] a set of similarity vectors are ranked and those in the middle (fuzzy) region are selected for manual labelling. The labelled instances are further used to train a classification model.

An interactive training model based on AL is proposed in [30]. In this approach all the record pairs are clustered by their similarity vectors. A number of similarity vectors from each cluster are selected and their corresponding pairs of records are selected for manual labelling. If all selected pairs of records from a cluster are labelled as match, then the remaining similarity vectors from the cluster are automatically labelled as match and added to the training dataset. Similarly, if all selected pairs of records from a cluster are labelled as non-match, the remaining similarity vectors in the cluster are automatically labelled as non-match and included in the training dataset. If the selected pairs of records from a cluster belong to different classes then the cluster is further split into sub-clusters. The process is repeated recursively until all the pairs of records are added to the training dataset.

A different semi-supervised learning based approach to RL is proposed in [31]. The system takes as input a small set of training examples, referred to as seeds, to initially train the classification model. The initial classification model

is then applied to classify the unseen data. A small percentage of the most confidently classified record pairs are selected to iteratively train the classification model. The process is repeated for a number of iterations or until all the unseen data are labelled. To maximize the performance of the model on unseen data an ensemble technique referred to as boosting is employed together with weighted Majority Voting. Random Forest and Multilayer Perceptrons are applied as the BCs in the ensemble. The proposed system requires a number of parameters to be specified, including the maximum number of iterations, the percentages of the most confident matching and non-matching examples to be selected in each iteration.

Semi-supervised learning significantly reduces the number of manually labelled examples required for generating a classification model. However, it still requires a certain amount of human input in the training process. In [9] two unsupervised RL methods were proposed. The first one, referred to as Clustering Record Linkage Model, uses k-means clustering to divide all similarity vectors into three clusters. Depending on the values in the similarity vectors, each cluster is labelled with one of the three matching status, match, non-match and possibly match. The similarity vectors in the first two clusters were automatically labelled, while the third cluster required manual labelling. With the second method, referred to as Hybrid Record Linkage Model, the RL process consists of two steps. First, clustering is used to predict the matching status of a small set of similarity vectors. In the second step, the labelled similarity vectors are used as training set for a supervised learning algorithm.

In [12] an unsupervised approach to RL based on automatic self-learning is proposed. In this work an approach to automatic seed selection referred to as nearest based was applied. With the nearest based approach, all similarity vectors are sorted by their distances (e.g., Manhattan or Euclidean distance) from the origin $[0, 0, \dots]$ and from the vector with all similarities equal to 1 ($[1, 1, \dots]$). Following this, the respective nearest vectors are selected as match and non-match seeds. The numbers of match and non-match seeds to be selected are taken as input parameters to the method. The authors evaluated three sizes

of the non-match seeds, namely 1%, 5% and 10% of the entire dataset. Following this, an appropriate ratio of similarity vectors was selected as match seeds. After selecting seeds, a classification model is trained incrementally as with the self-learning technique. In the same work it was observed that this approach did not provide good quality match seeds when the dataset contained only a small number of matching records or there were some fields with a large proportion of abbreviations.

In more recent work unsupervised techniques for RL based on maximizing the value of pseudo F-measure [32] were proposed [32], [33]. Pseudo F-measure is formulated based on the assumption that while different records often represent the same entity in different repositories, distinct record within one dataset is expected to denote distinct entity. It can be calculated using sets of unlabelled records. The idea is to find the decision rule for record matching which maximizes the value of the objective function (pseudo F-measure) applying genetic programming [32] or a hierarchical grid search [33]. The RL rules are formulated by manipulating weights and similarity measures for pairs of attributes, modifying the similarity threshold value and changing the aggregation function for individual similarities. To the best of our knowledge, the approaches based on the application of the pseudo F-measure are the state-of-the-art in the area of unsupervised RL.

It can be noted that, apart from the method presented in [12], [32], [33], in each of the aforementioned methods some level of human input is required. These methods are not applicable in many real-world situations. In particular for privacy preserving RL, where the data is private and confidential [10], it may be impossible to label the data manually. The method proposed in [12] does not require any labelled data, however, it performs significantly worse than supervised methods. At the same time a recent study found that on real data the pseudo F-measure is often negatively correlated with the true F-measure [33], which raises concerns about whether currently defined pseudo F-measure can be successfully applied to predict the real accuracy.

3. Preliminary and Problem Formulation

Given two sets of records R_1 and R_2, RL is defined as a task of identifying pairs of records $(r_1, r_2) \in R_1 \times R_2$ that refer to the same entity (e.g., a person). The decision whether a pair of records represent a match is based on the similarity between the two records on each of their fields, which can be determined with a similarity measure.

Definition 3.1. *(Similarity measure) Given two records r_1 and r_2 represented by N fields $f_1, \ldots, f_N$, a similarity measure m quantifies similarity between the two records on one of the fields. It returns a numeric value ranging between 0 and 1, referred to as a similarity value. A similarity measure is represented as $m(r_1.f_i, r_2.f_i)$ where $r_1.f_i$ and $r_2.f_i$ represent the values of field f_i in records r_1 and r_2 respectively.*

Value 1 indicates that the pair of records have the exact same values on a filed according to the similarity measure. Value 0 indicates that there is no similarity between the filed values for the two records. An example similarity measure could be Edit distance which for given two strings (e.g., *surnames*) determines the minimum number of operations that are required to transform one string into the other. For a pair or records represented by N fields a combination of N different similarity measure can be applied to determine similarity between the records on each of the N fields.

Definition 3.2. *(Similarity measure scheme). Given two records r_1 and r_2 represented by N fields $f_1, \ldots, f_N$, a similarity measure scheme is defined as $S_c = m_1, \ldots, m_N$, where m_i represents a similarity measure used to measure the similarity between r_1 and r_2 on field f_i, for $i = 1, \ldots, N$.*

Following the application of a similarity measure scheme, for each pair of records with N fields we can construct a N-dimensional vector representing similarity values between the records on the fields.

Definition 3.3. *(Similarity vector). Given two records r_1 and r_2 represented by N fields $f_1, \ldots, f_N$ and a similarity measure scheme $S_c = m_1, \ldots, m_N$, a*

similarity vector for r_1 and r_2 is defined as:

$$\overrightarrow{(S_c(r1, r2))} =< m_1(r_1.f_1, r_2.f_1), \ldots, m_N(r_1.f_N, r_2.f_N) > \quad (1)$$

The aim of RL process is to classify each similarity vector into one of two classes: matches (M) and non-matches (U). The goal of our work is to develop a learning algorithm, which, given two sets of records R_1 and R_2 and a set of available similarity measures M, can be used to construct a model for classifying a pair of records as match or non-match. The classification model should be trained without requiring any manually labelled data as input.

The general idea of our proposed approach is to generate an ensemble of self-learning classifiers using different similarity measure schemes. Self-learning is an approach to semi-supervised learning where small amount of labelled data and large amount of unlabelled data are used for training.

Definition 3.4. *(Self-learning). Given training dataset $L = \{x_i, y_i\}_{i=1}^{l}$ (refereed to as seeds) and unlabelled dataset $U = \{x_j\}_{j=l+1}^{l+u}$ (usually $L \ll U$), the self-learning process aims to train a classifier h on L initially and then use the classifier to label all unlabelled data in U. Following this, the most confidently classified unlabelled examples are selected as new training examples and moved into L. The process is repeated until a stopping condition is met.*

One may notice a major difference between self-learning and the aforementioned active learning. Instead of choosing the most confident examples to be labelled by itself, active learning actively selects the most problematic examples from the unlabelled dataset and asks these examples to be manually labelled.

With the proposed approach a number of different similarity measure schemes are selected. Each of the selected similarity measure schemes is used to generate a different set of similarity vectors. Each set of similarity vectors is then used to generate a different self-learning classifier. To a certain extent, our ensemble method is based on a similar principle to Bagging. With Bagging, each BC is trained with different (randomly selected) subset of the training dataset. Instead of randomly selecting samples for the training set, we construct different

training datasets using a set of different similarity measure schemes. As it is the case in Bagging, with the proposed approach all BCs are trained with the same learning algorithm but with different training datasets.

4. Proposed Approach

Figure 1 shows our proposed RL framework. Two sets of unlabelled records are provided as input. The RL process is performed in six steps. As with a typical RL approach, blocking is the first step and it can be thought of as a pre-processing step. We investigated two techniques, a standard blocking method referred to as canopy clustering [5] and a recently proposed unsupervised blocking scheme learner for blocking [34]. Better reduction ratio of the record pairs was obtain with the latter technique therefore we used it in our paper. To our best knowledge, this is the state-of-the-art method for unsupervised blocking in RL. It is relatively easy to implement and performs well empirically. The blocking method is divided into two phases. First, a weakly labelled dataset is generated automatically. In the second phase the problem of learning a blocking scheme from the weakly labelled dataset is cast as a Fisher feature selection problem [35].

The second step of the RL process is the selection of similarity measure schemes. In this step we search the whole space of all possible similarity measure schemes in order to select the most diverse subset of it. In the third step (seed selection with field weighting) each of the selected similarity measure schemes is first used to generate a set of similarity vectors. Then the automatic seed selection process is performed on each set of similarity vectors. As the output of this step different sets of seeds are selected. In the fourth step (Selecting the most diverse sets of seeds), the diversity between sets of seeds is measured using the proposed technique referred to as Seed Q Statistics. Only those most diverse sets of seeds are selected. In the fifth step the self-learning algorithm is applied with each of the selected sets of seeds. In the last step the proposed contribution ratios of BCs are used to eliminate the weakest BCs from the final

ensemble. Finally, for each pair of records the mode of the predictions of the selected self-learning models is provided as the final prediction. Each of the aforementioned steps is described in detail in the following sections.

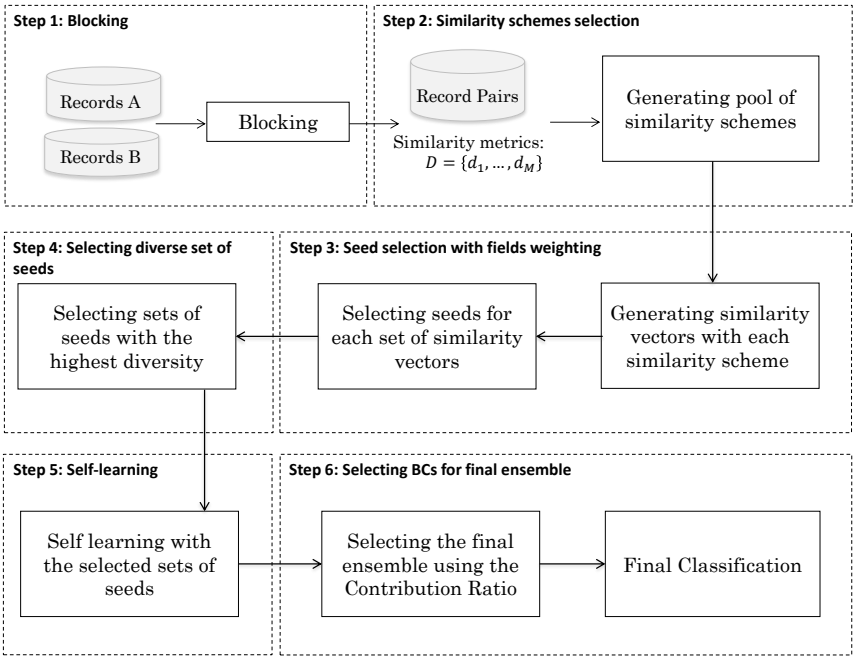

Figure 1: RL process with the proposed approach.

4.1. Selecting similarity measure schemes

For a given set of M similarity measures we could construct M^N possible similarity measure schemes of size N, where N is the number of fields. However, when M and N are large numbers, it would be infeasible to learn all BCs using each of the possible similarity measure schemes. In addition, the key of ensemble learning is to combine a set of diverse BCs. We need to select those similarity measure schemes that produce the most diverse sets of similarity vectors from the given dataset. We propose a method for selecting such a pool of similarity measure schemes. We first select a set of similarity measure for each of the fields. A pool of similarity measure schemes can then be generated, as a cross

product of the sets of similarity measure selected for each of the fields. In order to select a set of similarity measure for field f, in which no similarity measure is too similar to another, we define the similarity between two similarity measure.

Definition 4.1. *(Field similarity between records) Given a similarity measure m and a pair of records r_1 and r_2, each with N fields, $f_1, \ldots, f_N$, the field similarity between r_1 and r_2 on field f_i, for $i = 1, 2, \ldots, N$, is defined as: $m(r1.f_i, r2.f_i)$.*

Definition 4.2. *(Similarity between Similarity Measures) For a given set of records R and two similarity measures m_1 and m_2, let $\overrightarrow{m_1(R,f)}$ and $\overrightarrow{m_2(R,f)}$ be two vectors with each corresponding pair of elements in $\overrightarrow{m_1(R,f)}$ and $\overrightarrow{m_2(R,f)}$ representing the field similarity between each possible pair of records in R on field f. The similarity between m_i and m_j on field f is defined as:*

$$sim_f(m_i, m_j) = cossim(\overrightarrow{m_i(R,f)}, \overrightarrow{m_j(R,f)}) \tag{2}$$

Cosine similarity [36] measures the similarity between two vectors of an inner product space by the cosine of the angle between them. We aim to select a set of similarity measures for each of the fields, in which no *cossim* between two similarity measures is greater than a threshold.

The selection method is described in Algorithm 1. It takes as input a set of similarity measures, a set of records and parameter p. The parameter indicates the similarity threshold, i.e., the maximum similarity that two measures selected for the same field can have. The method selects a subset of similarity measures $\check{V}_f$ for each field f. The set V_f refers to all the similarity measures provided as input. In the first step a pair of similarity measures from V_f with the lowest similarity on f is selected and moved to $\check{V}_f$ (line 3). The remaining of the process is performed in iterations. First, we remove all similarity measures from V_f that have similarity on f higher than p with any of the similarity measures in $\check{V}_f$ (Line 5). Following this, in lines 6-7 we select a similarity measures from V_f which has the highest similarity on f with any of the measures from $\check{V}_f$ and move it to $\check{V}_f$. The process repeats until there is no similarity measures in V_f. In the

last step, a pool of similarity measure schemes is generated as a cross product of the similarity measures selected for each of the fields (line 10). The value of parameter p influences the number of similarity measures selected for each field and consequently the number of similarity measure schemes selected. The higher the value of p, the more similarity measure schemes will be generated. For $p = 1$, the number of similarity measure schemes will be M^N. If the run time of each similarity measure is bounded above by $O(c)$ then for P pairs of records represented by N fields the process of generating similarity vectors with one scheme should take $O(N \times P \times c)$. With our proposed approach the run time will be $O(s \times N \times P \times c)$ where s is the number of selected similarity schemes.

Algorithm 1 Generating a pool of similarity measure schemes

Input: $\psi = m_1, \ldots, m_M$: a set of similarity measures, R: a set of records each with N fields $f_1, \ldots, f_N$, $p \in (0, 1)$: similarity threshold

Output: $\check{C}$: a set of similarity measures schemes

1: **for all** $f \in F$ **do**
2: $V_f = m_1, \ldots, m_M$, $\check{V}_f = \emptyset$
3: Select a pair of similarity measures from V_f with the lowest similarity and move them to $\check{V}_f$
4: **while** $V_f \neq \emptyset$ **do**
5: Remove every m from V_f such that $\text{argmax}_{m_i \in \check{V}_f} sim_f(m_i, m) > p$
6: Select $m \in V_f$ where $m = \text{argmax}_{m_j \in V_f} \text{argmax}_{m_i \in \check{V}_f} sim_f(m_i, m_j)$
7: Move m from V_f to $\check{V}_f$
8: **end while**
9: **end for**
10: **return** $\check{C} = \check{V}_1 \times \cdots \times \check{V}_N$

Theorem 4.1. *Let ψ be a set of similarity measures and $\ddot{C}$ be a set of all possible similarity measure schemes generated from ψ for a set of records R. If $\check{C}$ is a set of similarity measure schemes selected by Algorithm 1, then the*

following is true:

$$\forall_{C_i \in \ddot{C}} \quad \exists_{C_j \in \check{C}} \quad \forall_{f_n \in F} \quad sim_{f_n}(m_n^i, m_n^j) \geq p \tag{3}$$

where $C_i = m_1^i, \ldots, m_n^i$, $C_j = m_1^j, \ldots, m_n^j$ *and* p *is the similarity threshold given as an input.*

Theorem 4.1 claims that the selection method presented in Algorithm 1 generates a pool of similarity measure schemes that covers a space of similarity measure schemes with the level of similarity equal to p. This means that for any similarity measure scheme C_i we can find a similarity measure scheme C_j from the selected pool, such that any corresponding pair of m_i and m_j from C_i and C_j have similarity higher than $p(sim(C_i, C_j) > p)$.

Proof. Following Algorithm 1 $\check{C}$ is generated as a cross product $\check{C} = \check{V}_1 \times, \ldots, \times \check{V}_N$. Let $C_i =< m_1^i, \ldots, m_N^i >$ be a similarity measure scheme selected from $\ddot{C}$. We show that there exists $C_j \in \check{C}$ such that $sim(C_i, C_j) > p$. We consider the following two scenarios:

I. $C_i \in \check{C}$ then $\forall_{f_n \in F}\ sim_n(m_n^i, m_n^j) = 1 > p$

II. $C_i \in \check{C}$

Let $C_j \in \check{C}$ be as follows:

$$C_j =< m_1^j, \ldots, m_N^j >: m_n^j = \begin{cases} m_n^i & \text{if } m_n^i \in \check{V}_n \\ \text{argmax}_{m \in \check{V}_n}\, sim_{f_n} & \text{if } m_n^i \notin \check{V}_n \end{cases}$$

If $m_n^i \in \check{V}_n$ then $sim_n(m_n^i, m_n^j) = 1 > p$. Otherwise we have:

$$m_n^i \notin \check{V}_n \implies \exists_{m \in \check{V}_n}\ sim(m, m_n^i) > p \implies \max_{m \in \check{V}_n}\ sim(m, m_n^i) > p$$

Finally we have the following.

$$\forall_{f_n \in F}\ sim_{f_n}(m_n^i, m_n^j) > p$$

□

4.2. Seed selection with field weighting

Each of the selected similarity measure schemes is used to generate a set of similarity vectors for all the pairs of records produced by the blocking algorithm. For each vector set, a small group of similarity vectors are automatically labelled as match and non-match, which will be used as seeds in the self-learning method.

4.2.1. Field weighting

In this paper we propose an improvement to the approach, proposed in [12], to automatic seed selection for self-learning. In the original approach in [12] those similarity vectors that are the nearest to the perfect match and perfect non-match as the seeds.

Definition 4.3. *(Perfect Match). A perfect match is a similarity vector, with each of its elements having a similarity value of 1, that is,*

$$\overrightarrow{1} =< x_i >_{i=1,\ldots,n}, \forall i\ x_i = 1 \tag{4}$$

which indicates that the pair of records represented by the similarity vector has the same value on each of their corresponding fields.

Definition 4.4. 5 *(Perfect Non-Match). A perfect non-match is a similarity vector, with each of its elements having a similarity value of 0, that is,*

$$\overrightarrow{0} =< x_i >_{i=1,\ldots,n}, \forall i\ x_i = 0 \tag{5}$$

which indicates that the pair of records represented by the similarity vector has completely different values on each of their corresponding fields.

In our improved approach we still select similarity vectors nearest to the perfect match and perfect non-match. However, we take into account the distinguishing power of fields when calculating the distance between a similarity vector and the perfect match or non-match. It is known that some fields may be more distinguishing than the others in the RL process [37]. For example, the field *last name* is more distinguishing than the field $first\ name$ given that more people share the first name than the last name. This is referred to as

the distinguishing power of a field. We say that a field has high distinguishing power if the similarity between any pair of records on the field is close to 1 when the records match and 0 otherwise.

Definition 4.5. *(Distinguishing Power of a Field) For a set of similarity vectors labelled as match (X^M) and non-match (X^U), the distinguishing power of field f_j is defined as:*

$$dp_{f_j} = \frac{\sum_{x \in X^M} x_j + \sum_{x \in X^U} (1 - x_j)}{|X^M| + |X^U|} \tag{6}$$

where x_j represents the value of field j in similarity vector x.

The numerator sums up the distances between each similarity vector from X^M and $\overrightarrow{0}$ and each similarity vector from X^U and $\overrightarrow{1}$ on f_j. The dp_{f_j} is the average of such distances among all vectors in X^M or X^U. It equals to 1 if all the matches have a similarity value of 1 on f_j, and all the non-matches have a similarity value of 0 on f_j. In such a case, field f_j has the highest distinguishing power for the given dataset.

Our intention is to assign a weight to each field in proportion to its distinguishing power. The weights are then used for calculating the distance between a similarity vector and either the perfect match or the perfect non-match in the seed selection process. Given that our proposed method is fully unsupervised and we do not have any labelled data we are not able to calculate the distinguishing power of a field as described in Definition 4.5. For this reason we instead use an unsupervised field weighting method that was proposed for the k-means clustering algorithm in [38]. We are able to show (Theorem 4.2) that the weight assigned to a field with this method is in proportion to the distinguishing power of the field.

The method in [38] first clusters the unlabelled records using the k-means algorithm. Then, a weight is assigned to each field based on the sum of within cluster distances of the field. Each of the within cluster distances for a field represents the difference on this field between the centroid of the cluster and one of the records in the cluster. The principle is to assign a higher weight to

a field with a lower sum of the within cluster distances. The weights are then used in the clustering process in the next iterations. The process iterates until there is no change to any of the weights. It has been shown in [38] that the weights of fields reflect their levels of significance.

In our work we follow the same principle as in [38] with two specific clusters representing two sets of seeds, one containing the match seeds (X^M with the perfect match as the centroid) and another containing the non-match seeds (X^U with the perfect non-match as the centroid). After the two sets of seeds are selected the weights of fields are calculated as follows:

$$\omega_j = \begin{cases} \frac{1}{|\{f_i : D_i = 0\}|} & \text{,if } D_j = 0, \\ 0 & \text{,if } D_j \neq 0 \wedge |\{f_{i \neq j} : D_i = 0\}| \neq 0, \\ \frac{1}{\sum_{k=1}^{n} (D_j / D_k)} & \text{,otherwise.} \end{cases} \tag{7}$$

The D_j is the sum of all the distances between every vector in X^M and the perfect match, and every vector in X^U and the perfect non-match on field f_j for $j = 1 \ldots n$. It can be seen from Definition 4.5 that, if $D_j = 0$ then the distinguishing power of f_j is 1. Therefore, if there are any fields with $D_j = 0$ (condition 1) they are assigned with equal weights and the remaining fields have a weight of 0 (condition 2) and are not taken under consideration while calculating the final similarity. If there is no field with $D_j = 0$ then all the fields are assigned with weights calculated using formula in condition 3. The weights total to 1 (Theorem 4.3).

Theorem 4.2 states that the weight of a field calculated with Formula 7 is proportional to the distinguishing power of the field.

Theorem 4.2. *For a set of similarity vectors, each labelled as either match (X^M) or non-match (X^U), for any two fields f_i and f_j we have:*

$$\omega_i \leq \omega_j \Leftrightarrow dp_{f_i} < dp_{f_j} \tag{8}$$

Proof. First we show that the theorem holds if $|\{f_i : D_i = 0\}| \neq 0$. According to the Formula 7, if there is one or more field with $D_i = 0$ then weight of any

field f_j is equal to either 0 (if $D_j \neq 0$) or $\frac{1}{|\{f_i:D_i=0\}|}$ (if $D_j = 0$). Obviously $0 < \frac{1}{|\{f_i:D_i=0\}|}$ and according to Definition 4.5, fields with $D_j = 0$ have the highest distinguishing power. Therefore: $dp_{f_j:D_j\neq 0} < dp_{f_i:D_i=0}$ so the theorem holds. Below we demonstrate that the theorem also holds when $|\{f_i : D_i = 0\}| = 0$.

$$\omega_i < \omega_j \Rightarrow \frac{1}{\sum_{k=1}^{n} D_i/D_k} < \frac{1}{\sum_{k=1}^{n} D_j/D_k} \Rightarrow \sum_{k=1}^{n} D_i/D_k > \sum_{k=1}^{n} D_j/D_k \Rightarrow$$

$$D_i > D_j \Rightarrow \sum_{x\in X^M}(1-x_i) + \sum_{x\in X^U} x_i > \sum_{x\in X^M}(1-x_j) + \sum_{x\in X^U} x_j \Rightarrow$$

$$\sum_{x\in X^M} 1 - \sum_{x\in X^M} x_i + \sum_{x\in X^U} x_i > \sum_{x\in X^M} 1 - \sum_{x\in X^M} x_j + \sum_{x\in X^U} x_j \Rightarrow$$

$$-\sum_{x\in X^M} x_i + \sum_{x\in X^U} x_i > -\sum_{x\in X^M} x_j + \sum_{x\in X^U} x_j \Rightarrow$$

$$-\sum_{x\in X^U} 1 - \sum_{x\in X^M} x_i + \sum_{x\in X^U} x_i > -\sum_{x\in X^U} 1 - \sum_{x\in X^M} x_j + \sum_{x\in X^U} x_j \Rightarrow$$

$$\sum_{x\in X^U} 1 + \sum_{x\in X^M} x_i - \sum_{x\in X^U} x_i < \sum_{x\in X^U} 1 + \sum_{x\in X^M} x_j - \sum_{x\in X^U} x_j \Rightarrow$$

$$\sum_{x\in X^M} x_i - \sum_{x\in X^U}(1-x_i) < \sum_{x\in X^M} x_j - \sum_{x\in X^U}(1-x_j) \Rightarrow$$

$$\frac{\sum_{x\in X^M} x_i - \sum_{x\in X^U}(1-x_i)}{|X^M \cup X^U|} < \frac{\sum_{x\in X^M} x_j - \sum_{x\in X^U}(1-x_j)}{|X^M \cup X^U|} \Rightarrow dp_{f_i} < dpf_j$$

The implication $dp_{f_i} < dp_{f_j} \rightarrow \omega_i < \omega_j$ can be proved accordingly. □

Theorem 4.3. $\sum_{i=0}^{n} \omega_i = 1$.

Proof. If there is at least one field f_i such as $D_i = 0$ then the weights of all the files with $D_i = 0$ are the same and add up to 1 so the theorem holds. Let's assume that there are N fields $f_1, \ldots, f_N$ and $\forall_{j=1\ldots,N} D_j \neq 0$. According to Formula 7 we have:

$$\omega_j = \frac{1}{\sum_{k=1}^{N} D_j/D_k} = \frac{1}{D_j/D_1 + \cdots + D_j/D_N} \tag{9}$$

Denoting $D_1 \times \cdots \times D_{i-1} \times D_{i+1} \times \cdots \times D_N$ by $\ddot{D}_{-i}$ and bringing all the fractions to the common denominator we get:

$$\frac{1}{\frac{D_j}{D_1} + \cdots + \frac{D_j}{D_N}} = \frac{\prod_{i=1}^{N} D_i}{\prod_{i=1}^{N} D_i + D_j \times \sum_{i=1\ldots n, i\neq j} \ddot{D}_{-i}} = \\ = \frac{D_j \times \ddot{D}_{-j}}{D_j \times \ddot{D}_{-j} + D_j \times \sum_{i=1\ldots N, i\neq j} \ddot{D}_{-i}} = \\ = \frac{\ddot{D}_{-j}}{\ddot{D}_{-j} + \sum_{i=1\ldots N, i\neq j} \ddot{D}_{-i}} = \frac{\ddot{D}_{-j}}{\sum_{i=1\ldots n} \ddot{D}_{-i}}$$

Consequently:

$$\sum_{j=1}^{N} \omega_j = \frac{\ddot{D}_{-1}}{\sum_{i=1}^{N} \ddot{D}_{-i}} + \cdots + \frac{\ddot{D}_{-N}}{\sum_{i=1}^{N} \ddot{D}_{-i}} = \frac{\ddot{D}_{-1} + \cdots + \ddot{D}_{-N}}{\sum_{i=1}^{N} \ddot{D}_{-i}} = 1$$

□

4.2.2. Seed Selection

For a given set of similarity vectors X, the seed selection process is performed in two phases as illustrated in Figure 2. All the steps of the seed selection proces are described in Algorithm 2. In the first phase, an initial set of similarity vectors is selected using the given distance metric (d_w) and two thresholds, t_M for match seeds and t_U for non-match seeds (lines 4-11). In this paper we use Manhattan distance. However, other distance metrics could also be used. Initially both Manhattan and Euclidean distances were considered but empirically Manhattan distance worked slightly better. At the beginning all the field weights have the same values. The minimum numbers of similarity vectors to be selected initially as match and non-match seeds are specified by two parameters, m_M and m_U (line 4 and 8). All similarity vectors within t_M distance from the perfect match and t_U distance from the perfect non-match are selected (lines 6 and 10). Initially, both thresholds, t_M and t_U are set to 0.05. They are gradually increased until the minimum numbers of matches and non-matches are reached (lines 5 and 9). After the initial set of seeds is selected, the process moves to Phase II, where the weights of fields are calculated using

Formula 7 and the seeds selected in Phase I (line 12). The new weights are then used to select a new set of seeds (lines 14-16).

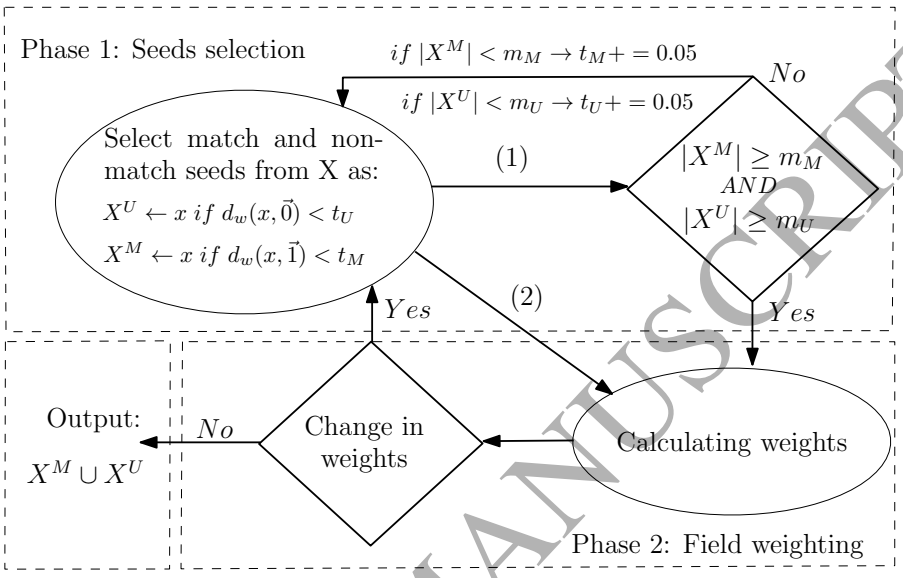

Figure 2: The complete process of automatic seed selection. In Phase 1 a small set of matches and non-matches is selected with equal weight assigned to each field. In Phase 2 the seeds selected in Phase 1 are used to calculate the weights of the fields. Phase 1 is then repeated using the new weights while calculating distances.

The computational complexity of the field weighting process is comparable to the k-means clustering algorithm for $k = 2$, which is $O(2 \times P \times i \times N)$ with i being the number of iterations. For s number of similarity schemes selected in the previous step the computational complexity is $O(s \times 2 \times P \times i \times N)$.

Example 1. To illustrate the seed selection process an example of selecting non-match seeds is shown in Figure 3. The example refers to a dataset with two fields, which means that each similarity vector contains two values (x and y) only. The similarity threshold t_U is set as 0.1. In the first iteration both fields have the same weight of 0.5. Consequently, based on the Manhattan distance metric, all those similarity vectors meeting the selection criterion $\frac{1}{2}|x-0|+\frac{1}{2}|y-0| < 0.1$ are selected as non-match seeds (blue dots). In the second iteration, based on the

Algorithm 2 Automatic selection of seeds

Input: X: set of unlabelled similarity vectors, m_M, m_U: minimum numbers of labelled matches and non-matches to be selected in Phase I

Output: X^M, X^U: sets of labelled match and non-match seeds

1: $X^M, X^U \leftarrow \Theta$
2: $\omega_{i=1...N} = \frac{1}{|number\ of\ attributes|}$
3: $t_M, t_U = 0$
4: **while** $|X^M| < m_M$ **do**
5: Increase t_M by 0.05
6: Get vectors from X within t_M distance from $\vec{0}$ and add them to X^M
7: **end while**
8: **while** $|X^U| < m_U$ **do**
9: Increase t_U by 0.05
10: Get vectors from X within t_U distance from $\vec{0}$ and add them to X^U
11: **end while**
12: Calculate new weights ω_i' using X^M and X^U and Formula 7
13: **while** $|\omega_i - \omega_i'| > \epsilon$ **do**
14: Remove all vectors from X^M and X^U
15: Select x from X within t_M weighted distance from $\vec{1}$ and add it to X^M
16: Select x from X within t_U weighted distance from $\vec{0}$ and add it to X^U
17: Calculate new weights using X^M and X^U
18: **end while**
19: **return** X^M, X^U

selected seeds, the weights are recalculated as $\frac{2}{3}$ and $\frac{1}{3}$, and the selection criterion is changed to $\frac{2}{3}|x| + \frac{1}{3}|y| < 0.1$. Consequently a new set of seeds is selected (red dots). In the third iteration, using the newly selected seeds, the weights are updated to $\frac{3}{4}$ and $\frac{1}{4}$ and the selection criterion is set as $\frac{3}{4}|x| + \frac{1}{4}|y| < 0.1$. In this case, no new seed is selected. Hence there is no change to the weights and no more iteration.

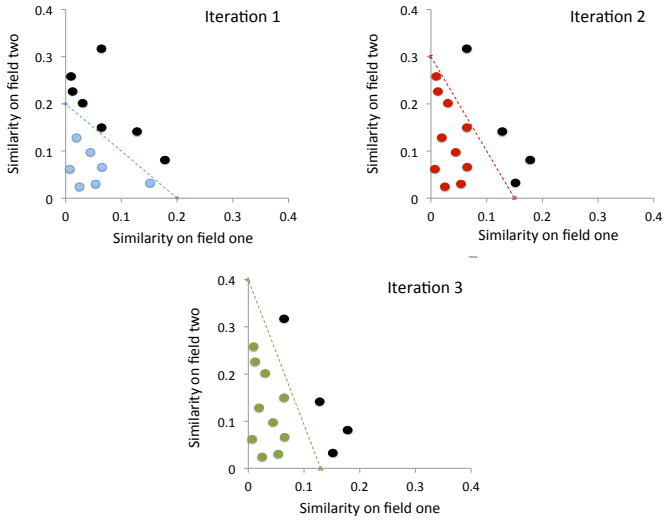

Figure 3: An example of the non-match seed selection process with 2-dimentional similarity vectors. The process is performed in 3 iterations. In the first iteration (blue) the weights of the fields are $(\frac{1}{2}, \frac{1}{2})$. In the second iteration (red) the weights were updated to $(\frac{2}{3}, \frac{1}{3})$. In the last iteration (green) the weights were set as $(\frac{3}{4}, \frac{1}{4})$.

4.3. Selecting highly diverse sets of seeds

For each set of similarity vectors we now have a small number of labelled examples (seeds). Consequently, any binary classifier can be trained using the selected seeds to classify the remaining unlabelled similarity vectors. With ensemble learning it is a common practice to select a collection of BCs that has the highest diversity to form an ensemble. Table 2 shows a simplified example of two ensembles, each consisting of three BCs. The output of each classifier

is denoted as 1 for being *correct* and 0 for being *wrong*. The final prediction of the ensemble is determined as the mode of the three individual predictions. It can be seen that in Ensemble I each of the BCs makes mistakes on different examples. As a consequence the combined classification is better than any of the BCs. In Ensemble II, every BC misclassified the same example. Therefore, combining them does not make any overall improvement.

Table 2: Two ensembles of three BCs. Each BC in Ensemble I makes a mistake on a different example, while every BC in Ensemble II makes a mistake on the same example.

Ensemble I				Ensemble II			
	x_1	x_2	x_3		x_1	x_2	x_3
C_1	0	1	1	C_1	0	1	1
C_2	1	0	1	C_1	0	1	1
C_3	1	1	0	C_1	0	1	1
$Final$:	1	1	1	$Final$:	0	1	1

In [23] different pairwise and non-pairwise diversity measures for classifier ensembles are presented. However, these diversity measures can only be calculated with labelled data. Given that our approach is fully unsupervised, we propose a diversity measure that does not require any labelled data. In [23] the Q statistic measure is recommended as the most appropriate diversity measure. The Q statistic is a pairwise diversity measure that is defined based on a 2×2 table representing the relationship between predictions of two classifiers as shown in Table 3.

Table 3: A 2×2 table representing agreements between predictions of two classifiers..

	C_1 is right	C_1 is wrong
C_2 is right	N^{00}	N^{10}
C_2 is wrong	N^{01}	N^{11}

The Q statistic for classifier C_1 and C_2, $Q_{1,2}$, is then defined as:

$$Q_{1,2} = \frac{N^{00}N^{11} - N^{01}N^{10}}{N^{00}N^{11} + N^{01}N^{10}} \tag{10}$$

The value of $Q_{1,2}$ is between -1 and 1. The classifiers with high values of

N^{00} and N^{11} (classifiers that make the same predictions) will have a positive value. On the other hand, the classifiers with high values of N^{10} and N^{01} (classifiers that have different classifications on the same examples) will have a negative value. For a set of L classifiers, the average Q statistic over every pair of classifiers is:

$$Q_{av} = \frac{2}{L(L-1)} \sum_{i=0}^{L-1} \sum_{j=i+1}^{L} Q_{i,j} \tag{11}$$

Therefore, we will select the set of classifiers with the lowest possible value of Q_{av}. We propose to use the Q statistic to select the most diverse sets of seeds. As a result, the final ensemble contains a set of self-learning models that were generated using the most diverse sets of seeds. Our observation is that the most diverse sets of seeds lead to the most diverse classifiers. We formalize seed Q statistic in Definition 4.6.

Definition 4.6. *(Seed Q statistics) For K sets of seeds $S_1, \ldots, S_K$, let $S = S_1 \cup \cdots \cup S_K$, Q statistics for two seed collections S_i and S_j, is defined as follows:*

$$Q_{S_{i,j}} = \frac{S^{00}S^{11} - S^{01}S^{10}}{S^{00}S^{11} + S^{01}S^{10}} \tag{12}$$

where:

$$S^{00} = x \in S : x \notin S_i \wedge x \notin S_j$$
$$S^{11} = x \in S : x \in S_i \wedge x \in S_j$$
$$S^{01} = x \in S : x \notin S_i \wedge x \in S_j$$
$$S^{10} = x \in S : x \in S_i \wedge x \notin S_j$$

If S_i and S_j are the same then we expect to obtain two same self-learning models, which means there will be no diversity between them ($Q_{s_{i,j}} = 1$). At the same time, if $S_i \cap S_j = \emptyset$ then the initial classifiers in the self-learning process will be trained on two completely different datasets. So we expect to have high diversity between the two models ($Qs_{s_{i,j}} = -1$). Similar to the standard Q statistic, we will select the ensemble with the lowest possible average value Qs_{s_av}. The

process of selecting the most diverse sets of seeds is presented in Algorithm 3. The time complexity of the seed selection process is $O(b \times (K - T))$ where b refers to the complexity of the evaluation function which is linear with respect to K and N.

Algorithm 3 Selecting similarity measure schemes for an ensemble

Input: $\check{S} = S_1, \ldots, S_K$: pool of seed collections, T: number of sets of seeds to be selected

Output: Sets of seeds with the lowest $Q_{s_a v}$

1: Calculate pair Q_s statistic matrix using $\check{S}$ and formula in Definition 4.6
2: **while** $|\check{S}| > T$ **do**
3: Select S from $\check{S}$ such as: $\text{argmin}_{s \in \check{S}}, \check{S}\, Q_{S_{av}}(\check{S} \neg S)$
4: Remove S from $\check{S}$
5: **end while**
6: **Return** $\check{S}$

4.4. Self-learning ensemble

The self-learning process is performed with each selected set of seeds. In [12] a SVM is used in the self-learning process. However, the computational complexity of training the SVM classifier may be a limitation when dealing with large amount of data. Ensemble methods require to train multiple BCs, which makes the problem even worse. In addition, for a large dataset the self-learning process may involve multiple iterations, which means that the SVM needs to be trained multiple times. To improve the efficiency of the learning process we apply the Stochastic Gradient Descent (SGD) algorithm for estimating the parameters of SVMs, which is very effective for large-scale online learning problems. SGD has been recently used [39] for the online training of various linear models. In general SGD is preferred for being faster as it optimizes parameters by using one training example at a time till it converges, instead of using the whole training dataset in each of the iterations. In the self-learning process the instances that the classifier is the most confident about are iteratively added to

the training dataset. With the proposed method the class probability distribution of an instance produced by the SVM-SGD algorithm as output is used to determine the level of its confidence on the classification of the instance. For example, if the class probability distribution of an instance is 0 on match and 1 on non-match then the SVM-SGD classifies the instance as non-match and the level of its confidence on the classification is 1. The complete classification process performed through the self-learning of a SVM-SGD classifier is described in Algorithm 4. In the first step the seeds are used to generate the initial SVM-SGD classification model (line 3). The initial trained classification model is then used to classify the remaining unlabelled instances (line 4). Instances that have been classified with a level of confidence above the threshold are added to the training dataset (lines 6-10). This process iterates until all instances are added to the training dataset.

4.5. Selecting the final ensemble using the contribution ratios of BCs

Following the self-learning process, a collection of classification models is generated. Since the proposed method is fully unsupervised we are not able to evaluate how good each of classification models is. Therefore, there is a risk of including classifiers with very poor accuracy (i.e., below 0.5) which are not valid in general, into the ensemble. In order to address this issue we propose a statistic which takes into account the contribution ratio of each individual BC to the final output of the ensemble. Each BC makes a prediction on each record pair as match or non-match. Following this, the mode of all the predictions by all the BCs is taken as the prediction of the ensemble. The contribution ratio of a BC is formalized in 4.7.

Definition 4.7. *(Contribution Ratios of Base Classifiers) Let* $\check{C} = C_1, \ldots, C_n$ *be an ensemble of BCs,* $X = x_1, \ldots, x_m$ *be a set of unlabelled examples on which any two BCs in* $\check{C}$ *have different prediction. The contribution ratio of* C_i

Algorithm 4 self-learning SVM-SGD

Input: X: set of unlabelled similarity vectors, X^M: set of match seeds, X^U: set of non-match seeds, p: minimum number of unlabelled similarity vectors selected in each iteration

Output: X^{MU}: completely labelled dataset

1: $X^{MU} \leftarrow X^M \cup X^U$
2: **while** $X = \emptyset$ **do**
3: Update SVM-SGD classifier on X^{MU}
4: Classify X with SVM-SGD
5: $S = \Theta, T = 1$
6: **while** $|S| < p$ **do**
7: $S \leftarrow$ select examples from X witch confidence at least T
8: Remove S from X
9: Decrease T byt 0.05
10: **end while**
11: Add S into X^{MU}
12: **end while**
13: **Return** X^{MU}

is defined as:

$$contr(C_i) = \frac{1}{m}\sum_{j=1}^{m}\begin{cases}1 & if\ C_i(x_j) = \check{C}(x_j)\\ 0 & otherwise\end{cases} \tag{13}$$

where $C_i(x_j)$ and $\check{C}(x_j)$ represent the predictions of classifier C_i and the ensemble on x_j respectively.

Our intuition is that if there is a BC with very poor accuracy in the ensemble, its contribution ratio will be significantly lower than the other BCs. Removing this BC from the ensemble may help improve the performance of the ensemble. Therefore, we calculate the average contribution ratio of all BCs and only those BCs with contribution ratios above the average are included in the final ensemble. In the last step, the mode of the predictions of the selected BCs is taken as the final output for each candidate record pair.

5. Experimental evaluation

In this section we present the experimental evaluation of our proposed RL approach. The objectives of our experimental evaluation are:

1. To evaluate whether the proposed automatic seed selection with field weighting can improve the quality of the seeds in the self-learning process.
2. To evaluate whether the proposed ensemble learning technique can improve the overall performance of individual self-learning models and hence alleviate the problem of manually selecting the best similarity measure scheme.
3. To evaluate whether the proposed Seed Q Statistic and the Contribution Ratio help form a better ensemble.
4. To evaluate whether the proposed unsupervised approach to RL can outperform recently proposed semi-supervised and unsupervised RL methods and achieve comparable results to supervised classification methods such as J48 and SVM-SGD.

The proposed self-learning SVM-SGD ensemble method has been implemented in Java with the Weka and Simmetrics Java libraries for SVM-SGD classification and string similarity measures respectively. Five commonly used similarity measures for RL have been used, namely Jaro (J), Smith-Waterman (SW), Q-Gram (Q), Jaro-Winkler (JW) and Levenshtein edit distance (L). However, any other similarity measures could be applied.

The experiments were conducted with four datasets commonly used for evaluating RL methods: Restaurant[1], Cora[1], ACM-DBLP[2] and DBLP-Scholar[2]. The Restaurant dataset contains 864 restaurant records (372,816 record pairs with 112 pairs of matching records), each with five fields, including name, address, city, phone and type. The Cora dataset is a collection of 1,295 (837,865 record pairs with 17,184 pairs of matching records) citations to computer science papers. Each citation is represented by 4 fields (author, title, venue, year). The ACM-DBLP and DBLP-Scholar are a bibliographic datasets of Computer Science bibliography records represented by four attributes. The total number of entity pairs (cross product) is 6,001,104 and 168,181,505 for ACM-DBLP and DBLP-Scholar respectively.

5.1. Automatic Seed Selection

We first evaluated whether the proposed technique for field weighting for seed selection improves the performance of the self-learning algorithm. The algorithm for automatic seed selection takes two parameters as input, m_M and m_U, which specify the minimum numbers of match and non-match seeds that need to be selected in the first phase of the seed selection process. We used $m_U = 1\%$ of the total number of similarity vectors and a smaller value of $m_M = 0.01\%$ due to the imbalanced distribution of match and non-match examples. Commonly in RL the number of matching pairs of records is significantly smaller than the number of non-matches. Therefore, the value of m_M needs to be smaller. If we

[1] `https://www.cs.utexas.edu/users/ml/riddle/data.html`

[2] `http://dbs.uni-leipzig.de/en/research/projects/object_matching/fever/benchmark_datasets_for_entity_resolution`

set it too high we could get a large number of false positive seeds. Since the number of non-matches is much larger than the one of matches we can allow for m_U to be much higher to provide a bigger set of seeds.

We compared the precision and recall of the seeds selected in two phases. Phase I is where the seeds are selected using unweighed Manhattan distance. Phase II is where the seeds selected in Phase 1 are used to calculate the weights of fields and then the final seeds are selected using the weighted Manhattan distance. The results are presented in Tables 4-7. Due to the large initial pool of the similarity measure schemes generated in Step 2, we only compared the results for the 10 most diverse sets of seeds selected in Step 4. The second column in each of the tables indicates which similarity measure scheme was selected for generating the ensemble with each of the datasets.

Table 4: Precision and Recall of the seeds selected for the Restaurant dataset without (Phase I) and with (Phase II) field weighting

		Phase I				Phase II			
		Precision		Recall		Precision		Recall	
BC	Sim. Scheme	M	NM	M	NM	M	NM	M	NM
1	J + J +J +J +J	0.91	1	0.13	0.01	0.86	1	0.45	0.02
2	J + J +J +J +Q	0.96	1	0.18	0.01	0.18	1	0.2	0.3
3	J + Q +J +J +J	0.95	1	0.16	0.01	0.93	1	0.13	0.04
4	Q + J +J +J +Q	0.95	1	0.36	0.01	1	1	0.73	0.9
5	Q + J +J +Q +J	0.98	1	0.36	0.1	1	1	0.73	0.89
6	Q + J +J +Q+Q	0.94	1	0.16	0.02	1	1	0.73	0.9
7	Q + Q+J +J +J	0.92	1	0.3	0.01	1	1	0.74	0.91
8	Q + Q +J +J +Q	0.97	1	0.1	0.02	1	1	0.73	0.9
9	Q + Q +J +Q +J	0.97	1	0.27	0.03	1	1	0.72	0.89
10	Q + Q +J +Q +Q	0.92	1	0.3	0.05	1	1	0.73	0.91

With the Restaurant dataset, in three out of ten cases (cases 1-3) better precisions of the match seeds were obtained in Phase I. In the other cases better results for match seeds were obtained in Phase II. For the non-match seed significantly better results were obtained in Phase II. For the Cora dataset, apart from cases 6-8, better precision and recall of both match and non-match seeds were obtained in Phase II. For the DBLP-ACM dataset, apart from case 2-3

Table 5: Precision and Recall of the seeds selected for the Cora dataset without (Phase I) and with (Phase II) field weighting

		Phase I				Phase II			
		Precision		Recall		Precision		Recall	
BC	Sim. Scheme	M	NM	M	NM	M	NM	M	NM
1	J + J +J +J	0.99	0.98	0.14	0.01	0.99	0.99	0.28	0.3
2	J + J +L +J	0.92	0.98	0.14	0.01	0.99	1	0.28	0.32
3	J + SW +J +J	0.99	0.99	0.16	0.01	0.99	1	0.28	0.32
4	J + SW +J +SW	0.99	0.99	0.16	0.02	0.99	1	0.25	0.39
5	J + SW +L +SW	0.99	0.99	0.14	0.01	0.99	0.99	0.23	0.3
6	J + SW +SW +J	0.99	0.99	0.16	0.02	0.92	0.89	0.14	0.3
7	J + SW +SW +SW	0.99	0.99	0.15	0.02	0.98	0.83	0.25	0.32
8	SW + SW +J +J	0.99	0.99	0.15	0.02	0.99	0.81	0.2	0.35
9	SW + SW +J +SW	0.99	0.99	0.15	0.02	0.99	1	0.29	0.39
10	SW + SW +L +J	0.99	0.99	0.16	0.01	0.99	0.99	0.23	0.32

Table 6: Precision and Recall of the seeds selected for the DBLP-ACM dataset without (Phase I) and with (Phase II) field weighting

		Phase I				Phase II			
		Precision		Recall		Precision		Recall	
BC	Sim. Scheme	M	NM	M	NM	M	NM	M	NM
1	J + J +J +J	0.73	0.99	0.1	0.63	0.73	0.99	0.1	0.63
2	J + L +J +J	0.72	1	0.09	0.73	0.67	0.98	0.27	0.9
3	J + SW +J +J	0.66	0.99	0.09	0.74	0.36	0.99	0.29	0.87
4	J + Q +J +J	0.75	0.99	0.11	0.79	0.9	0.99	0.43	0.9
5	L + Q +J +J	0.79	0.99	0.12	0.9	0.86	0.99	0.4	0.93
6	SW + J +J +J	0.86	0.99	0.4	0.9	0.91	0.99	0.35	0.93
7	SW + Q +J +J	0.84	0.99	0.4	0.93	0.86	0.99	0.42	0.93
8	Q + J +J +J	0.73	0.99	0.1	0.83	0.9	0.99	0.2	0.93
9	Q + SW +J +J	0.72	0.99	0.1	0.83	0.89	0.99	0.26	0.93
10	Q+Q+J+J	0.84	0.99	0.4	0.9	0.94	0.99	0.27	0.93

(M precision) and 6, 10 (M Recall) better match and non-match seeds were obtained in Phase II. For the DBLP-Scholar dataset the better results in term of recall were obtained in Phase II for both matches and non-matches. However, much better precision for matches was obtained in Phase I. It can be observed from the results that in majority of the cases higher precision and higher recall, for both match and non-match seeds, were achieved in Phase II for each of the

Table 7: Precision and Recall of the seeds selected for the DBLP-Scholar dataset without (Phase I) and with (Phase II) field weighting

		Phase I				Phase II			
		Precision		Recall		Precision		Recall	
BC	Sim. Scheme	M	NM	M	NM	M	NM	M	NM
1	L+L+J+J	1	0.99	0.1	0.2	0.47	0.99	0.2	0.21
2	J+J+J+J	1	0.99	0.09	0.21	0.5	0.99	0.24	0.2
3	L+L+L+J	1	0.99	0.09	0.2	0.47	0.99	0.2	0.21
4	L+L+S+J	1	0.99	0.12	0.21	0.47	0.99	0.2	0.21
5	S+J+J+J	1	0.99	0.12	0.21	0.48	0.99	0.25	0.22
6	S+J+L+J	1	0.99	0.1	0.21	0.48	0.99	0.25	0.22
7	S+J+S+J	1	0.99	0.1	0.21	0.48	0.99	0.25	0.22
8	S+L+J+J	1	0.99	0.15	0.21	0.45	0.99	0.2	0.21
9	S+L+L+J	1	0.99	0.1	0.21	0.45	0.99	0.2	0.21
10	S+L+S+J	1	0.99	0.12	0.21	0.45	0.99	0.2	0.21

datasets apart from the DLBP-Scholar dataset. We presume that the low precision for DBLP-Scholar could be caused by the large number of missing values in this dataset. Because of the missing values the algorithm was not able to determine the weights correctly which caused a large number of false positive seeds. This issue will be addressed in the future work.

In order to see how the quality of the seeds affects the self-learning process, for each set of seeds from Tables 4-7 we evaluated the performance of the self-learning model. Each of the diagrams in Figure 4 shows the F-measure obtained by the self-learning models for each of the 4 datasets. For each of the datasets, the self-learning process was performed with the similarity vectors generated with each of the 10 selected similarity schemes. The self-learning process was performed with field weighting (we refer to this method as SL-AW) and without field weighting (we refer to this method as SL). Note that the self-learning without field weighting is the same as the method proposed and evaluated in [12]. It can be noted that for the Cora dataset the SL-AW performed better than SL for each of the similarity schemes apart from cases 6-8, which is in line with the results presented in Table 5. For the Restaurant dataset, the SL method outperformed SL-AW in one case (2). It can be observed from the corresponding

row in Table 4 that for the same similarity measure scheme significantly better precision of match seeds was obtained with the SL method. At the same time, it can be noted that even though for similarity measure schemes 1 and 3 better match seeds were selected with SL, the SL-AW method still performed better in term of F-measure. This could be due to the fact that the precision and recall of the non-match seeds were slightly better with the SL-AW in those two cases. For the DBLP-ACM dataset the SL method performed equally or slightly better than SL-AW in 4 cases (1, 3, 6 and 10). It can be observed from the corresponding rows in Table 6 that for each of the four similarity measure schemes SL obtained either better precision or recall of the match seeds. It can be observed from Figure 4 that even though for the DBLP-Scholar dataset the SL method had much higher precision that SL-AW, it performed better only in four cases (4, 5, 8, 10). The reason for this could be the higher recall for the matches obtained by SL-AW.

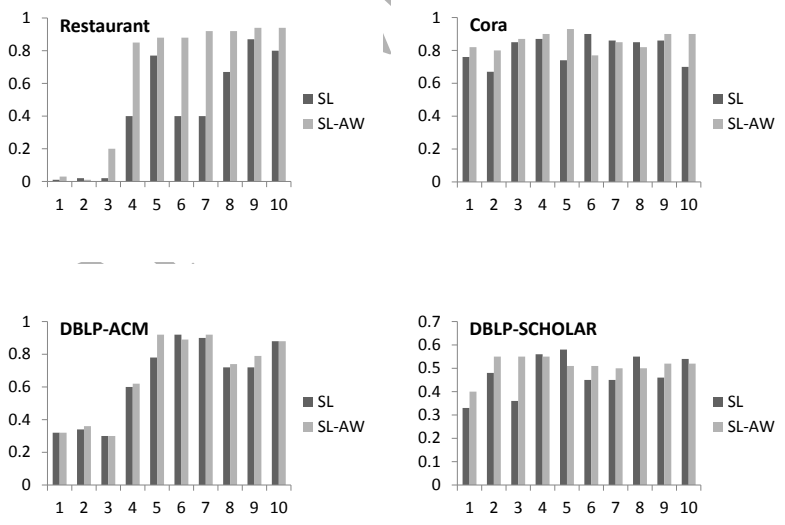

Figure 4: F-measures obtained by the self-learning models with (SL-AW) and without (SL) field weighting. Each chart refers to one dataset. For each dataset the 10 similarity measure schemes selected for the ensemble were evaluated

To further explore the relation between seeds and the final output of the self-learning process we measured the correlation between the precision and recall of the seeds and the F-measure obtained by the self-learning model. Following our analysis, it appeared that the most correlated with the F-measure are the recall of the non-match seeds and the precision of the match seeds. This observation reflects the experimental results shown in Figure 4.

Based on the results shown in Tables 4-7 and Figure 4 it can be concluded that the proposed field weighting technique allows obtaining better quality of seeds and consequently improves the final performance of the self-learning algorithm.

5.2. Ensemble Selection

The process of creating the ensemble starts with creating a pool of similarity measure schemes. For each of the 4 datasets we used the same similarity threshold (0.8) for selecting similarity measure schemes. The threshold was selected empirically. We noted that for the value of 0.8 the initial pool of similarity measure schemes (and BCs) was big enough and manageable. For any smaller value we received only 3 or 4 similarity schemes, which wasn't enough to generate an ensemble. The size of the initial pool for each dataset is shown in Table 8. For the Restaurant and Cora datasets 24 different similarity measure schemes were generated, for the DBLP-ACM and DBLP-Scholar it was 16 and 17 respectively. Each of the similarity measure schemes was then used for generating a set of similarity vectors. Following this and the automatic seed selection, a group of 10 most diverse sets of seeds were selected. Finally, following the self-learning process, the final ensemble was selected based on the contribution ratios of the BCs in the ensemble. The last row in Table 8 shows the size of the final ensemble for each of the datasets.

Table 9 shows the contribution ratio of each BC in the ensemble. We can see that for each of the 4 datasets some of the BCs have a significantly lower contribution ratio than others. From Figure 4 we can see that for each of the datasets the same classifier obtained significantly lower F-measure than the

Table 8: Initial and final sizes of the ensemble for each of the 4 datasets.

	Datasets			
	Restaurant	Cora	DBLP-ACM	DBLP-Scholar
Sim. measure schemes	24	24	16	17
Final ensemble size	8	8	8	7

Table 9: Contribution ratio of each of the 10 BCs in the ensemble. Each of the columns refers to one BC in an ensemble. For each of the datasets, each of the classifiers was generated with different similarity scheme as shown in Tables 4-7

BC:	BC_1	BC_2	BC_3	BC_4	BC_5	BC_6	BC_7	BC_8	BC_9	BC_{10}
Restaurant	**0.3**	**0.3**	0.9	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Cora	0.98	**0.89**	0.98	0.97	0.99	**0.59**	0.97	0.98	0.99	0.97
DBLP-ACM	**0.32**	**0.65**	0.88	0.95	0.95	0.96	0.95	0.95	0.94	0.97
DBLP-Sch.	**0.36**	0.89	0.89	0.89	0.9	**0.74**	**0.71**	0.93	0.97	0.97

other BCs. For the Restaurant dataset, for example, the first two BCs have a lower contribution ratio than the average, which is reflected in the results in Figure 4. The results in Figure 4 and Table 9 have shown that the contribution ratio is a good indicator of BCs with poor accuracy. The bold cells in Table 9 indicate which of the BCs were removed from the final ensemble.

5.3. Record Pair Classification

In this section we evaluate the classification performance of the proposed ensemble method. As a baseline for the proposed approached we used the following methods.

Semi-supervised Boosted Classifiers [31]. This is recently proposed minimally supervised approach which uses both ensemble learning and self-learning. It requires a seed training set as input which is then used to start the self-learning process. The seed set is composed of a set of similarity vectors labelled as matches and a set of similarity vectors labelled as non-matches. In each iteration of the learning process the AdaBoost algorithm is used to first train BCs with the seed training set and then classify unlabelled similarity vectors representing candidate pairs of records. A small number of similarity vectors classified with

the greatest confidence are added to the training set. The algorithm runs for a predefined number of iterations. We implemented the algorithm as described in [31] with the same seed training set as selected in our approach. Given that in our approach different seeds were selected for different similarity schemes we evaluated the boosting algorithm with each set of seeds individually and then presented the best result.

Pseudo F-measure [32]. This RL algorithm was implemented as described in the original paper. The algorithm takes as input two sets of records, two sets of potential attributes and a set of similarity measures. We used the same five similarity measures as in our approach . Following this a genetic algorithm is applied with the pseudo F-measure as the fitness function. The output of the algorithm is a RL rule composed of selected pairs of attributes, their weights, similarity measure for each pair of attributes, aggregation function and similarity threshold. All the parameters including rates for combination operators, rates for mutation options and termination criterion were set to the same values as described in the original paper. The methods based on the pseudo F-measure have a great advantage over any other approach since they don't require any training data. However, as demonstrated in [33] formulation of the right pseudo F-measure is still an unsolved problem.

In addition we compared our method with a two supervised classification models, J48 decision tree and the SGW model.

Due to the imbalanced distribution of matches and non-matches among record pairs [9] the classification accuracy is not a suitable measure for the evaluation of a RL approach. In this paper we use the F-measure instead of accuracy, which is the harmonic mean of Recall and Precision, and is most commonly used. For the evaluation of the supervised techniques we used 10-fold cross validation, while the RL process with the semi-supervised and unsupervised methods was performed on the whole dataset. Table 10 shows the results of all the classification methods we have used in this paper. The first 2 rows refer to 2 supervised methods, J48 decision tree and SVM-SGD. For training the supervised models we generated a set of labelled similarity vectors using each

of the same 10 similarity measure schemes that were selected for the ensemble. Each supervised learning method was evaluated with each of the training sets and the best results were recorded. Rows 3 and 4 refer to the Semi-supervised Boosted Classifier and the pseudo F-measure respectively. The last 4 rows in Table 10 show the results obtained by the following methods.

- Best SL-AW: self-learning model with the proposed field weighting for seed selection. For each dataset, the self-learning process was performed with each set of similarity vectors and the best result was recorded,
- SL-AW-E: ensemble composed of all SL-AW models,
- SL-AW-D-E: ensemble composed of 10 SL-AW models selected using the proposed seed diversity measure,
- SL-AW-D-A-E: ensemble composed of the SL-AW models selected using the seed diversity measure and the average contribution ratio (the final ensemble).

To measure the statistical significance of the obtained results we performed the McNemar's statistical test [40]. The F-measure values in bold in Table 10 indicate that the difference between the corresponding method and SL-AW-D-E was statistically significant with the p values equals to 0.05.

It can be seen that the final ensemble performed equally well as the best of the self-learning models (Best SL-AW) in term of F-measure. This confirms that the proposed method alleviates the issue of selecting the most appropriate similarity measure scheme in the absence of labelled data. Comparing the results of SL-AW-E and SL-AW-D-E it can be seen that the selection method based on the seed diversity significantly improved the performance of the ensemble for each of the 4 datasets. For each of the datasets the difference was statistically significant. We can also observe that SL-AW-E performed worse than the Best SL-AW. This shows that generating an ensemble using all of the similarity measure schemes without taking under consideration the diversity

Table 10: P-precision, R-recall and F-measure obtained by each of the evaluated classification methods on each of the 4 datasets.

Method	Datasets					
	Restaurant			Cora		
	P	R	F	P	R	F
Best J48	0.96	0.97	0.96	0.97	0.94	**0.95**
Best SVM-SGD	0.93	0.97	0.95	0.98	0.92	**0.95**
Semi-Sup SL	0.92	0.77	**0.84**	0.75	0.98	**0.85**
Pseudo F-measure	0.98	0.75	0.93	0.9	0.6	**0.72**
Best SL-AW	0.97	0.9	0.93	0.98	0.88	0.93
SL-AW-E	0.97	0.84	**0.9**	0.94	0.86	**0.9**
SL-AW-D-E	0.96	0.89	0.93	0.98	0.88	0.93
SL-AW-D-A-E	0.97	0.9	0.94	0.98	0.88	0.93
	DBLP-ACM			DBLP-Scholar		
	P	R	F	P	R	F
Best J48	0.94	0.98	**0.96**	0.69	0.6	**0.64**
Best SVM-SGD	0.95	0.98	**0.96**	0.58	0.74	**0.65**
Semi-Sup SL	0.85	0.96	**0.9**	0.45	0.74	**0.57**
Pseudo F-measure	0.94	0.98	**0.96**	0.36	0.86	**0.5**
Best SL-AW	0.88	0.96	0.92	0.43	0.75	0.55
SL-AW-E	0.9	0.84	**0.87**	0.4	0.81	**0.54**
SL-AW-D-E	0.88	0.92	0.9	0.44	0.75	0.55
SL-AW-D-A-E	0.87	0.97	**0.92**	0.44	0.75	0.55

does not provide the expected outcome. Comparing SL-AW-D-E and SL-AW-D-A-E we can see that the ensemble selection method based on the contribution ratio improved the final performance of the Restaurant and DBLP-ACM but the difference was statistically significant only for DBLP-ACM dataset. It did not make any difference for the two remaining datasets. This indicates that the weak BCs identified through the contribution ratio have a negative impact on the final prediction of the ensemble in DBLP-ACM dataset. For Restaurant, Cora and DBLP-Scholar outliers do not affect the performance of the ensemble.

The results in Table 10 demonstrate the ensemble obtained comparable results to the supervised classification methods with 3 out of 4 datasets. For the DBLP-Scholar our proposed approach performed significantly worse. However, it can be as well observed that the semi-supervised SL and pseudo F-measure

methods performed equally bad with this dataset. It may be the case that none of the three methods can handle dataset with significant amount of missing data. We would like to look into this problem in more detail in the future work.

It can be noted in Table 10 that the semi-supervised SL method performed significantly worse than the proposed approach with the Restaurant, Cora and DBLP-ACM datasets. In the original paper better results for this method are reported. However, this is most likely related to the fact that in the original version of the method manually labelled data are provided as seeds for the self-learning process. In our paper the method was evaluated with the same set of automatically labelled seeds that were applied with our proposed approach. Apart from this, in the original paper the authors applied 28 different similarity measures without any selection. In our paper, for a fair comparison, we applied the same 5 similarity measures that were used with our proposed approach.

The method based on maximizing the pseudo F-measure performed outstandingly well with the DBLP-ACM dataset, obtaining equal results with the supervised methods. Nevertheless, it performed significantly worse for Cora and DBLP-Scholar datasets. It has been discussed in [33] that the pseudo F-measure tend not to be correlated with the original F-measure for some real world datasets, which must have been the case in our experiments. No statistically significant difference between the pseudo F-measure and our proposed method was noted for the Restaurant dataset. A number of interesting observations have been made based on the presented results. First, as shown in Figure 4, SL and SL-AW can obtain significantly different results for different similarity measure schemes. This indicates that the performance of both methods relies on the selection of an appropriate similarity measure scheme. However, it can be seen that the ensemble method always performs equally well as the best of the individual classifiers. Therefore, the ensemble technique has the advantage of not relying on the selection of an appropriate similarity measure scheme. Second, using field weighting in the automatic seed selection process can provide better quality of the seeds and consequently lead to better classification. Finally, for 3 out of 4 datasets, the proposed method obtained comparable results

with the supervised methods. Our experimental results have shown that the proposed self-learning ensemble approach can be applied to the RL problem when no labelled data is available. The big advantage of the approach is that it alleviates the problem of manually selecting the best similarity measure scheme without labelled data.

5.4. Analysis of computational complexity

The runtime results are determined for a workstation with Intel(R) Core(TM) i7-4790 CPU @ 3,60 GHz processor, 16 GB (RAM) and 64-bit Windows 7 Operation System. For each of the datasets the proposed method was evaluated with the record pairs produced as output by the blocking method. Table 11 shows the number of record comparisons before and after blocking. Table 12

Table 11: Number of record pairs comparison before and after blocking

	Restaurant	Cora	DBLP-ACM	DBLP-Scholar
Total number of record pairs	372,816	837,865	6,001,104	168,772,783
Number of record pairs after blocking	50,853	181,850	554,726	3,675,685

shows the runtime of each of the steps in the proposed RL approach for each of the 4 datasets. It can be noted that for small datasets (Restaurant, Cora) each of the steps is performed very efficiently. For the last two largest datasets (DBLP-ACM and DBLP-Scholar) the steps of generating similarity schemes (I) and selecting seeds with the highest diversity (IV) still have low execution time. However, the execution time of the remaining 3 steps is much higher. For step II (generating similarity vectors) this was an expected output since the time complexity of a record linkage algorithm is dominated by the number of record comparison performed. This is because the performance bottleneck is usually the expensive comparison of attribute values between records [6]. The aim of blocking is the reduction of the number of record comparisons. While we received a good reduction ration for each of the datasets as a result of the blocking

process, we increased the number of record comparisons by applying different combination of similarity schemes to generate the ensemble. In the self-learning process (step V) a SVM-SGD is re-trained in each iteration with the available labelled data. It has been demonstrated that SVM-SGD is very efficient with large-scale learning [41]. Given the nature of ensemble learning and self-learning the training process of SVM-SGD needs to be repeated $s \times j$ times where j is the number of iterations.

It can be noted that the time complexity of each of the steps could be reduced by decresing the size of the ensemble, which is the number of similarity schemes (s) selected in step I. Optimization of the similarity measure schemes selection process will be considered in the future work.

Table 12: Execution time in seconds for each step of the proposed RL method. I: Generating similarity schemes, II: Generating similarity vectors, III: Selecting Seeds, IV: Selecting seeds with the highest diversity, V: self-learning.

	I	II	III	IV	V
Restaurant	0.6	50	1	0.5	26
Cora	3.2	85	1	0.5	150
DBLP-ACM	3	545	31	5	1159
DBLP-Scholar	2	1772	1050	548	1600

6. Conclusions

It this paper we have proposed a new unsupervised approach to RL, which combines ensemble learning with automatic self-learning and unsupervised field weighting. The ensemble is generated using a number of different similarity measures schemes. The initial pool of similarity measure schemes is selected using cosine similarity. Each of the selected similarity measure schemes is then used to generate a set of similarity vectors. An unsupervised field weighting technique has been proposed for seed selection to improve the self-learning process. Following the automatic seed selection, we use the proposed unsupervised ensemble selection method based on seed diversity. Each selected set of seeds is then used as input to the self-learning algorithm. The contribution ratio of

each BC has been used to select a set of BCs to form the final ensemble. The final prediction is obtained as mode of the predictions of self-learning models from the ensemble.

Our experimental results have shown that the proposed self-learning ensemble method can be successfully applied in RL when no labelled data is available. It is shown that applying unsupervised field weighting in the automatic selection of seeds improves the quality of the initial training dataset in the self-learning process and leads to better classification results. By applying an ensemble of self-learning models we are able to obtain as good results as the best of the individual models. The proposed approach is not able to outperform supervised classification models. However, it can obtain comparable results. In comparison to some existing unsupervised RL techniques our proposed approach seems to perform better overall.

There are two limitations that have been identified. First, the proposed approach cannot handle missing data very well. We presume that this is mainly related to the process of features weighting while selecting the seeds for self-learning. Second, the proposed approach requires a larger number of record pair comparisons because it uses multiple similarity schemes for generating the similarity vectors.

In future work we intend to investigate the problems related to the scalability of the proposed approach. We want to evaluate how the proposed approach performs with different numbers of selected similarity measure schemes. In this paper we only used 5 different similarity measures. However, we would like to increase the number of the similarity measures provided as input to the algorithm and then optimize the process of selecting similarity measure schemes so that fewer sets of similarity vectors need to be calculated.

References

[1] F. Naumann, M. Herschel, An introduction to duplicate detection, Synthesis Lectures on Data Management 2 (1) (2010) 1–87.

[2] P. Christen, Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection, Springer Science Business Media, 2012.

[3] A. K. Elmagarmid, P. G. Ipeirotis, V. S. Verykios, Duplicate record detection: A survey, IEEE Transactions on knowledge and data engineering 19 (1).

[4] W. Cohen, P. Ravikumar, S. Fienberg, A comparison of string metrics for matching names and records, in: Kdd workshop on data cleaning and object consolidation, Vol. 3, 2003, pp. 73–78.

[5] R. Baxter, P. Christen, T. Churches, et al., A comparison of fast blocking methods for record linkage, in: ACM SIGKDD, Vol. 3, Citeseer, 2003, pp. 25–27.

[6] R. C. Steorts, S. L. Ventura, M. Sadinle, S. E. Fienberg, A comparison of blocking methods for record linkage, in: International Conference on Privacy in Statistical Databases, Springer, 2014, pp. 253–268.

[7] J. Wang, G. Li, J. X. Yu, J. Feng, Entity matching: How similar is similar, Proceedings of the VLDB Endowment 4 (10) (2011) 622–633.

[8] R. Isele, C. Bizer, Learning expressive linkage rules using genetic programming, Proceedings of the VLDB Endowment 5 (11) (2012) 1638–1649.

[9] M. G. Elfeky, V. S. Verykios, A. K. Elmagarmid, Tailor: A record linkage toolbox, in: Data Engineering, 2002. Proceedings. 18th International Conference on, IEEE, 2002, pp. 17–28.

[10] D. Vatsalan, P. Christen, V. S. Verykios, A taxonomy of privacy-preserving record linkage techniques, Information Systems 38 (6) (2013) 946–969.

[11] T. G. Dietterich, Ensemble methods in machine learning, in: International workshop on multiple classifier systems, Springer, 2000, pp. 1–15.

[12] P. Christen, Automatic record linkage using seeded nearest neighbour and support vector machine classification, in: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2008, pp. 151–159.

[13] K. Nigam, R. Ghani, Analyzing the effectiveness and applicability of co-training, in: Proceedings of the ninth international conference on Information and knowledge management, ACM, 2000, pp. 86–93.

[14] J. Kittler, M. Hatef, R. P. Duin, J. Matas, On combining classifiers, IEEE transactions on pattern analysis and machine intelligence 20 (3) (1998) 226–239.

[15] X. Li, L. Wang, E. Sung, A study of adaboost with svm based weak learners, in: Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on, Vol. 1, IEEE, 2005, pp. 196–201.

[16] T. G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization, Machine learning 40 (2) (2000) 139–157.

[17] L. He, Q. Song, J. Shen, Z. Hai, Ensemble numeric prediction of nearest-neighbor learning, Information Technology Journal 9 (3) (2010) 535–544.

[18] G. Zenobi, P. Cunningham, Using diversity in preparing ensembles of classifiers based on different feature subsets to minimize generalization error, in: European Conference on Machine Learning, Springer, 2001, pp. 576–587.

[19] L. Breiman, Bagging predictors, Machine learning 24 (2) (1996) 123–140.

[20] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32.

[21] Y. Freund, R. Schapire, N. Abe, A short introduction to boosting, Journal-Japanese Society For Artificial Intelligence 14 (771-780) (1999) 1612.

[22] D. Ruta, B. Gabrys, Classifier selection for majority voting, Information fusion 6 (1) (2005) 63–81.

[23] L. I. Kuncheva, C. J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Machine learning 51 (2) (2003) 181–207.

[24] M. Bilenko, R. J. Mooney, Adaptive duplicate detection using learnable string similarity measures, in: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2003, pp. 39–48.

[25] X. L. Dong, D. Srivastava, Big data integration, in: Data Engineering (ICDE), 2013 IEEE 29th International Conference on, IEEE, 2013, pp. 1245–1248.

[26] V. Christophides, V. Efthymiou, K. Stefanidis, Entity resolution in the web of data, Synthesis Lectures on the Semantic Web 5 (3) (2015) 1–122.

[27] A. Arasu, M. Gotz, R. Kaushik, On active learning of record matching packages, in: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, ACM, 2010, pp. 783–794.

[28] S. Sarawagi, A. Bhamidipaty, Interactive deduplication using active learning, in: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2002, pp. 269–278.

[29] W. W. Cohen, J. Richman, Learning to match and cluster large high-dimensional data sets for data integration, in: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2002, pp. 475–480.

[30] Q. Wang, D. Vatsalan, P. Christen, Efficient interactive training selection for large-scale entity resolution, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2015, pp. 562–573.

[31] M. Kejriwal, D. P. Miranker, Semi-supervised instance matching using boosted classifiers, in: European Semantic Web Conference, Springer, 2015, pp. 388–402.

[32] A. Nikolov, M. dAquin, E. Motta, Unsupervised learning of link discovery configuration, in: Extended Semantic Web Conference, Springer, 2012, pp. 119–133.

[33] A.-C. N. Ngomo, K. Lyko, Unsupervised learning of link specifications: deterministic vs. non-deterministic, in: Proceedings of the 8th International Conference on Ontology Matching-Volume 1111, CEUR-WS. org, 2013, pp. 25–36.

[34] M. Kejriwal, D. P. Miranker, An unsupervised algorithm for learning blocking schemes, in: Data Mining (ICDM), 2013 IEEE 13th International Conference on, IEEE, 2013, pp. 340–349.

[35] Q. Gu, Z. Li, J. Han, Generalized fisher score for feature selection, arXiv preprint arXiv:1202.3725.

[36] C. Perone, Machine learning:: Cosine similarity for vector space models (part iii), Pyevolve. sourceforge. net/wordpress.

[37] M. G. de Carvalho, A. H. Laender, M. A. Goncalves, A. S. da Silva, A genetic programming approach to record deduplication, IEEE Transactions on Knowledge and Data Engineering 24 (3) (2012) 399–412.

[38] J. Z. Huang, M. K. Ng, H. Rong, Z. Li, Automated variable weighting in k-means type clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (5) (2005) 657–668.

[39] S. Shalev-Shwartz, Y. Singer, N. Srebro, Pegasos: Primal estimated sub-gradient solver for svm, in: Proceedings of the 24th international conference on Machine learning, ACM, 2007, pp. 807–814.

[40] B. Bostanci, E. Bostanci, An evaluation of classification algorithms using mc nemars test, in: Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012), Springer, 2013, pp. 15–26.

[41] A. K. Menon, Large-scale support vector machines: algorithms and theory, Research Exam, University of California, San Diego 117.